The Design and Evaluation of a Scalable Wireless MAC Protocol

Technical Report UIUCDCS-R-2006-2793

Chun-cheng Chen, Eunsoo Seo, and Haiyun Luo
Dept. of Computer Science
Univ. of Illinois at Urbana-Champaign
Urbana, IL 61801-2302
{chen35,eseo2}@uiuc.edu, haiyun@cs.uiuc.edu

ABSTRACT

Contention-based MAC protocols such as IEEE 802.11 DCF are known to perform well in a network with small number of nodes. As the number of nodes increases, the cost of collisions increases and dominates. This increasing collisions quickly drives every client in the network into wasted and unfair idleness, resulting in performance degradation. In this paper, we propose SMAC, a scalable MAC protocol, that preserves the advantages of 802.11 while avoiding its drawbacks. The design of SMAC leverages the efficient channel utilization of 802.11 DCF at low load and 802.11 PCF at high load. The SMAC-enabled AP controls the access of its clients by combining the contention and schedule-based access mechanisms and determining the optimal contention level among the clients. The SMAC-client simply indicates its intention of channel access. Through thorough analysis and evaluation, we show that SMAC preserves the channel utilization efficiency of 802.11 DCF, scales up to any number of nodes, and preserves perfect short-term and long-term fairness among the contending clients.

1. INTRODUCTION

As the IEEE 802.11 becomes the one dominating standard which is widely accepted for the wireless networking technology for Local Access Network (LAN), many of the laptops, PDAs, cell phones, and even desktop computers are equipped with IEEE 802.11 NICs when they are manufactured. Because of the sudden proliferation of IEEE 802.11 devices, the number of nodes which is covered by one Access Point (AP) has also been increased. In a very populated area such as universities, conference sited and downtowns of big cities, it is observed that more that one hundred users are connected to one AP [5]. Users in those sites usually experience very low throughput or even frequent disconnections. This is because the original IEEE 802.11 DCF (and PCF, as well) has not been designed with scalability in mind. This problem will be more severe in the future because IEEE 802.11 is being spreaded fast and more devices such as digital cameras [7] or cars which have not been considered to be connected to the networks began to be connected to the networks through IEEE 802.11.

Contention-based MAC protocols such as IEEE 802.11 DCF are known to perform well in a network with small number of nodes. As the number of nodes increases, the cost of collisions increases and dominates. On the contrary, polling-based MAC protocols such as IEEE 802.11 PCF work more efficiently with larger number of nodes [11]. However, the performance of polling-based MAC protocols can be degraded if there are many idle nodes in the network. We examine the problems of IEEE 802.11 DCF and PCF in section 2. Then we present how to combine the advantages of these two approaches: a MAC protocol which maintains the aggregate throughput as the number of nodes increases regardless of the ratio of the idle nodes in section 3.

For further optimization, we propose to use the locality of packet transmission. That is, the node which transmits a packet is likely to have more packets in its output queue. This property reduces the cost of polling significantly and increase the overall performance of the proposed MAC protocol.

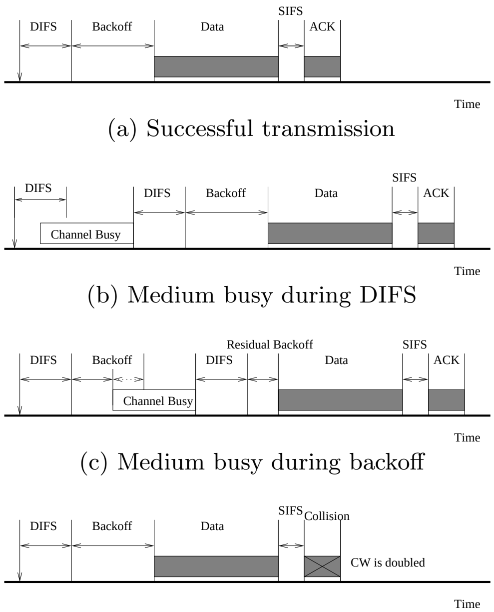
There are several techniques which may mitigate this problem such as spatial reuse by controlling the transmission power [6, 8], usage of multiple orthogonal channels [1] and admission control [5]. They can help solving the problem to some extent by reducing the number of concurrent contending nodes, but they are not scalable by themselves, either. We talk about these techniques in section 5.

2. THE SCALABILITY OF IEEE 802.11

In this section, we examine the IEEE 802.11 MAC protocols and show that they do not scale well as the number of nodes increases.

2.1 DCF

The Distributed Coordination Function (DCF) [10] is the mandatory MAC protocol of IEEE 802.11. It is based on carrier sense multiple access with collision avoidance (CSMA/CA), which uses random backoff to avoid collisions. Figure 1 depicts how 802.11 DCF works. When a node has a packet to send, it first senses the carrier. If the channel is idle, it first waits for DIFS. Then it chooses a random backoff time within the contention window (CW) and waits for the backoff time. If the channel is idle for the backoff time, it sends the packet, waits for SIFS and listens to the channel to receive ACK (figure 1(a)). If the channel becomes busy during the first DIFS time, the node waits until the channel becomes idle and re-starts to wait for DIFS (figure 1(b). If the channel becomes busy during the backoff



(a) Successful transmission

(b) Medium busy during DIFS

(c) Medium busy during backoff

(d) Collision

Figure 1: IEEE 802.11 DCF

time, it freezes the backoff timer immediately and it waits until the channel become idle. Then it first waits for DIFS and waits for the residual backoff timer (figure 1(c)). If the node does not receive any ACK after sending a packet or there is a collision during the reception of the ACK, it doubles the contention window (figure 1(d)). This mechanism is called *binary backoff*. A valid value of CW is in the range of $[CW_{min}, CW_{max}]$. When a node successfully transmits a packet and receives an ACK, it resets its CW to CW_{min} not to waste time with excessive backoff. When a node fails a packet transmission, it doubles CW (with upper limit of CW_{max}) to reduce collisions. If a node fails packet transmissions for seven times serially, it resets its CW to CW_{min} to prevent the long-term unfairness. Given the number of contending nodes, the optimal CW can be calculated [3]. Note that the necessary information is not the number of nodes but the number of *contending* nodes which have non-empty output buffers. However, it is impossible to know the number of contenders, so DCF resets/doubles the CW when it succeeds/fails a packet transmission instead of using a constant CW.

One question which arises here is whether the number two (which is used for *binary* backoff) is optimal or not. The exponential coefficient (two in binary backoff) determines how fast the CW reaches a *good* value which reflects the current congestion level. In that sense, if the network has a lot of contending nodes, it is better to use large exponential coefficient such as 3 or 4 than 2. Contrarily, if there are a small number of contending nodes in the network, smaller exponential coefficient such as 1.3 or 1.7 might be better for the performance. If the exponential coefficient is too small, it induces many collisions. If the exponential coefficient is too large, channel is under-utilized. As the number of nodes increases, the optimal CW also increases. Therefore DCF which uses constant exponential coefficient suffers from severe collisions as the number of nodes increases.

2.2 PCF

The other optional MAC protocol which IEEE 802.11 defines is the Point Coordination Function (PCF) [10] which can be used in conjunction with DCF. Contrary to DCF, PCF is a centralized solution.

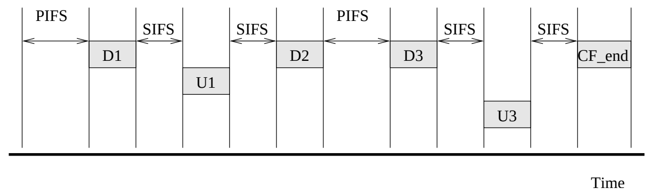


Figure 2: IEEE 802.11 PCF

Figure 2 shows how PCF works. When it starts, AP first waits for PIFS and sends a downstream data packet (D1) to the first node (n_1) in its network. If n_1 has an upstream packet, it sends the packet (U1) to the AP after SIFS. Then the AP moves on to the next node n_2 and sends another downstream packet (U2). If the polled node (n_2) does not have an upstream packet, it just does not do anything. Then AP detects it after PIFS, and continues polling other nodes immediately. When AP finishes polling, it sends a CF_{end} packet as an end marker. Note that there is no backoff until CF_{end} because AP controls the transmission schedule of all nodes.

If all nodes are backlogged, PCF is scalable unlike DCF whose frequency of collisions increase as the number of nodes increase. However in reality, a network has non-negligible portion of idle nodes. If it is the case, AP wastes some time which is proportional to the number of idle nodes for polling them, and it makes the overall performance not scalable.

3. DESIGN OF THE SCALABLE MAC

3.1 Scalable MAC

In SMAC, the clients in a single BSS are categorized into two groups–S and C. All clients in groups S must have frames to transmit and their transmissions are scheduled by the AP. The clients in group C are those who have no frames impending and access the channel through contention. Roughly speaking, the AP randomly chooses one of the clients in S to be scheduled for transmission. The clients in group C contend for the channel in between the transmissions of the scheduled clients.

When the AP receives the DATA from a client x in group S, it determines *who* is the next scheduled transmission client and *when* it should transmit by including $nextSTA$ and $nextCW$ in the ACK to x. All the other clients in the BSS must be able to overhear this ACK and decode the ACK frame. The client in S matching $nextSTA$ backs off at $nextCW$ slots; while other clients in S silently wait for their turns.

A client is initially in group C. When having DATA to transmit, it waits for the ACK containing the $nextSTA$ and $nextCW$, backs off randomly between zero and $nextCW-1$ slots. Since the next scheduled client backs off at $nextCW$ slots, it transmits DATA after all the clients in C finishes their contention. The contending clients indicate in the DATA whether they have more frames to send. When receiving the DATA from a contending client x with more frames to send, AP adds x to the set of scheduled clients if it is not there and replies with ACK. When receiving the ACK from the AP, the contending client with more frames to send changes its membership to S and starts waiting for its scheduled turn of transmission. If the contention fails due to collision, contending clients simply wait for the next ACK frame announcing $nextSTA$ and $nextCW$. Note that AP has all the information of the associated clients, it may depend on the number of clients in S and clients in C to adjust the opening of the contention period. Finally, since all the clients in S must have more frames to send, the AP removes the transmitter x from S if the received DATA indicates no more frames to send.

Regardless of the contending clients, SMAC is essentially a scheduling-based MAC protocol to avoid congestion-induced collision. This requires AP and the clients to have a consistent view of the membership. However, since wireless channel errors could occur in a fairly hight rate, the clients and AP may not always have the same view of the scheduled set. For example, a client x may transmit DATA indicating more frames to send, AP will add x into S. But the replied ACK may be lost, resulting x thinking itself is still in C. Similarly, a client x may transmit DATA asking for removal from S, AP removes x from S. But the replied ACK may again be lost, resulting that x thinks itself is still in C. For the former case, x will contend for the channel during the contention period and eventually be added to S consistently. For the latter case, we adopt a timer similar to 802.11 that is (re)triggered when a scheduled client x transmits DATA. When the timer expires without receiving the ACK, x changes itself into C and contend for the channel. This conservative approach guarantees that clients will never think themselves in S while in reality they are in C. But the reverse is not true...our solution is to...

Ideally, if all the clients belong to S, SMAC achieves the best possible total throughput since no time is spent on contention-induced collision and idleness. To account for potential clients' joining and leaving the scheduled set S, we leave some space in between the transmissions of two scheduled clients for other potential clients to join S. However, too much space would lead to wasted idle time slots while too little space leads to heavy collision of the contending clients. In the next section, we present the mechanism to obtain the optimal time slots to accommodate the contenting clients while preserving the fairness among the clients in the BSS.

3.2 Optimal Contention Window

The clients in C and client in S are *orthogonal* to each other in that they do not contend with each other. Nevertheless, the transmission of a scheduled client servers as a flag indicating the start and end of a contending epoch. Since no time slot is wasted among the clients in S, optimizing the overall throughput is equivalent to optimizing the throughput among the contending clients. Several people have proposed optimizing the contention-based throughput by tuning the contention window size [2], but they fall short of short-term fairness among the clients. Heusse et.al [3] proposed a new access mechanism based on having each node converging to the same contention window size. We adopt a similar approach to [3] but have the AP calculate the optimal contention window size $nextCW$ based on the number of collisions in the previous contention period and announce the result to be used by the clients in C in the next contention period.

Let P_e denote the channel attempt probability of the contending nodes, P_i denote the idle probability of the contending nodes, P_c denote the collision probability of the contending nodes, and P_t denote the successful channel transmission probability, N denote the number of contending nodes in C. We know that:

$$\begin{aligned} P_e &= \frac{2}{cw+1} && (1)\\ P_t &= NP_e(1-P_e)^{N-1} && (2)\\ P_i &= (1-P_e)^N && (3)\\ P_c &= 1-P_t-P_i \\ &= 1-NP_e(1-P_e)^{N-1}-(1-P_e)^N && (4)\end{aligned}$$

The throughput Thr of the contending nodes can be expressed as:

$$Thr = \frac{P_t E[P]}{P_t T_t + P_c T_c + P_i \sigma} \quad (5)$$

, where we assume the following variables are known: $E[P]$ is the expected DATA size, σ is aSlotTime, T_t and T_c are the average transmission duration and average collision duration respectively.

Substituting P_t, P_c, P_i into Eq (5), we obtain Thr as a function of the only unknown variable P_e. Thr is maximized when P_e is derived from the following equation:

$$1 - NP_e^{opt} = \eta(1-P_e^{opt})^N \quad (6)$$

,where $\eta = 1 - \frac{\sigma}{T_c} = \frac{67.17}{68.17}$ for 802.11.

Therefore, by estimating N, we can calculate P_e^{opt}, and use Eq(1) to find the optimal contention window size $nextCW$ to be used for the next round. It turns out the N can be easily estimated ($\hat{N}$) from the collision probability P_c^{prev} and channel attempt probability P_e^{prev} of the previous contention period. In short, let the number of collisions in the previous contention period be n_c. Let n_s be the number of successful transmissions in the previous contention period, and cw^{prev} be the previous optimal contention window size used. We obtain Eq (7):

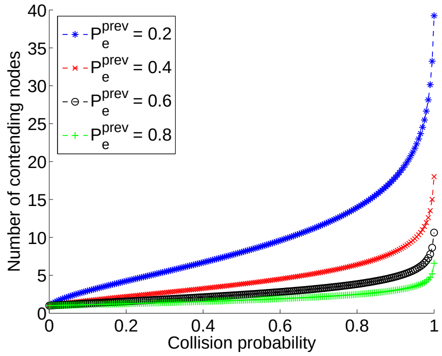
$$\frac{n_c}{cw^{prev}} = 1 - \hat{N}P_e^{prev}(1-P_e^{prev})^{\hat{N}-1} - (1-P_e^{prev})^{\hat{N}} \quad (7)$$

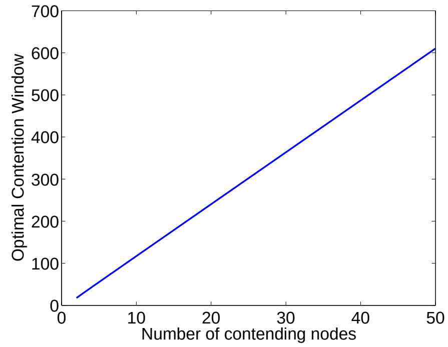
From Eq (7), we plot in Figure 3 the estimated number of contending stations ($\hat{N}$) based on the previous channel attempt probability ($P_e^p rev$) and the measured collision probability ($\frac{n_c}{cw^{prev}}$). We also plot the optimal contention window for different numbers of contending stations in Figure 4. Finally, obtaining $\hat{N}$ using Eq (7), we can use Eq (6) to obtain the $nextCW$ for AP to announce in the next contention period. The optimal contention window based on $P_e^p rev$ and $\frac{n_c}{cw^{prev}}$ is plotted in Figure 5.

The pseudo-code for the scalable MAC protocol is shown is Figure 6 and Figure 7.

3.3 Illustrative Examples

Suppose there are three clients A, B, and C in the BSS. Client A has been in the scheduled set of the AP. Client B and C have more frames to transmit, but they are not yet added to the scheduled set by the AP. Figure 8 shows one example of how B and C are added to the scheduled set S of the AP. The perceived channel channel status at





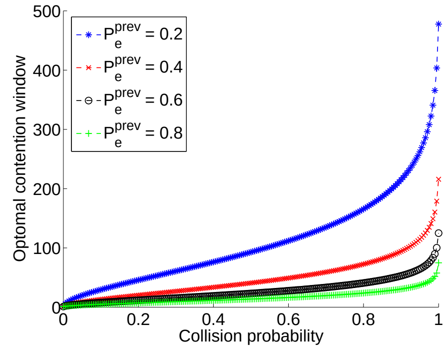


Figure 3: Number of contending stations for different channel attempt probabilities and collision probabilities

Figure 4: Optimal contention window for different numbers of contending stations

Figure 5: Optimal contention window for different channel attempt probabilities and collision probabilities

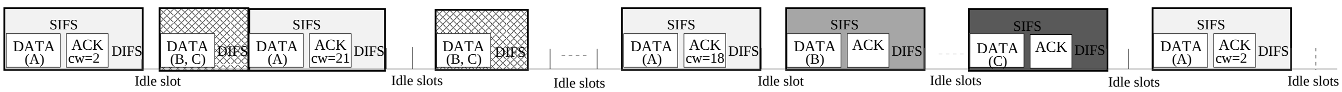


Figure 9: Frame exchange sequence and detailed timing of Figure 8

```
recvDATA(f, x)
f: the received DATA frame
x: the transmitter of the received frame
if x has more frames to send then
    insert x into S if x ∉ S
else
    remove x from S if x ∈ S
if x was in S already then
    ACK.nextCW ← calcOptCW()
    if S.size() > 0 then
        ACK.nextSTA ← randomly pick y ∈ S
    else
        ACK.nextSTA ← CONTEND
else
    if S.size() > 0 then
        if x is just inserted to S & S.size() = 1 then
            ACK.nextSTA ←x
            ACK.nextCW ← calcOptCW()
        else
            ACK.nextSTA ← WAIT
    else
        ACK.nextSTA ← CONTEND
        ACK.nextCW ← calcOptCW()
    send(ACK)
```

Figure 6: SMAC for AP

```
recvACK(f, a)
f: the received ACK frame
a: the transmitter of the received frame
if a = myAP & ContTimer is busy then
    stop ContTimer
if previously sent DATA indicates more frames to send then
    whichSet ← SCHEDULING
else
    whichSet ← CONTENDING
if whichSet = SCHEDULING then
    if I am f.nextSTA then
        backoff at f.nextCW
else
    if f.nextSTA ≠ WAIT then
        backoff randomly in [0, f.nextCW)
overhearACK(f, a)
f: overheard ACK frame
a: the transmitter of the overheard frame
if a = myAP & ContTimer is busy then
    stop ContTimer
if a = myAP & f.nextSTA ≠ WAIT then
    if send_timer is busy then
        stop send_timer
        if whichSet = SCHEDULING then
            whichSet = CONTENDING
    if whichSet = CONTENDING then
        if I have frames to send then
            if backoff timer is busy then
                stop backoff timer
            backoff randomly in [0, f.nextCW)
    else
        if I am f.nextSTA then
            backoff at f.nextCW
```

Figure 7: SMAC for Client

the AP can be classified into (1) successful transmission, (2) collision, and (3) channel idle. Since scheduled nodes do not contend with contending nodes, only the contending nodes may result in collision. Thus A's transmission is always successful.

Suppose the initial contention window is set to 2 in the ACK from AP to A, after overhearing this ACK B and C each randomly backs off in [0, 2) aSlotTime and A backs off at 2 aSlotTime. With only 2 slots, the probability of collision between B and C is 50%. To illustrate the optimal contention window calculation, let's suppose B and C both choose to transmit the data after 1 aSlotTime, resulting in data collision. Now, the AP measured that there is a collision during the contention period of length 2 slots. Since the previous announced contention window value is 2, the collision probability is $P_c = 1/2 = 0.5$, previous channel attempt probability $P_e^{prev} = 2/(2+1) = 2/3$. Substituting P_c and P_e^{prev} into Eq (7), we obtain $\hat{N}$ to be 2, which maps to 20 as the optimal contention window. AP then announces this value in the ACK to client A. Again A will back off at 20 aSlotTime, B and C back off in [0, 20). Although the probability for B and C's transmission to collide again drops to $1/20 = 0.5$, let's suppose B and C collide again. Following the same process, we obtain the optimal contention window size to be 43. In fact, if B and C keeps colliding with

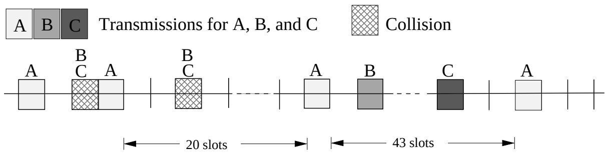


Figure 8: Perceived channel status at the AP for one scheduled client A and two contending clients B and C

each, the optimal contention window converges at around 85, shown in Figure 10. Note that the optimal contention window is obtained by measuring the collision probability at the AP, contending nodes do not need to perform the expensive binary exponential backoff which often leads to wasted idle channel. Furthermore, it adjusts the contention window faster than the traditional binary exponential backoff when there is a sudden congestion and converges to a much smaller value when continuous collisions happen.

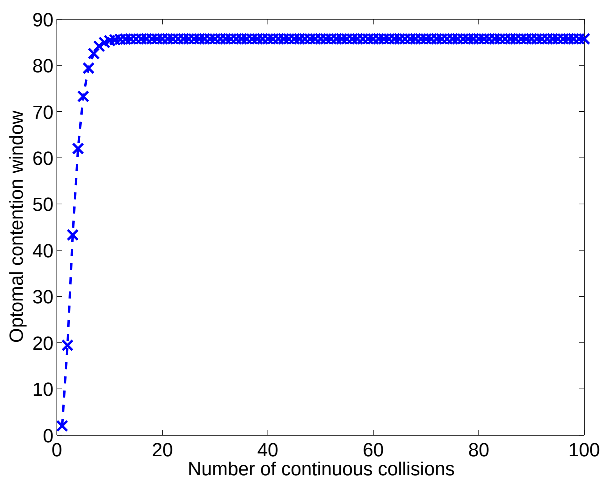


Figure 10: Optimal contention window evolution when B and C in Figure 8 keeps colliding

4. PERFORMANCE EVALUATION

We implement SMAC in *ns-2* simulator version *2.29*. For the current ns-2 802.11 implementation, nodes receive packets only when the RSS from the sender is greater than the receiving threshold, and the impact of any signal with RSS less than the carrier sense threshold is completely ignored - no matter how many those signals are. This is obviously an over-simplification of the reality. We replace this part of 802.11 functions with the ones developed in [4], so that all signals from the simulation are taken into account at receiver, and the combined signal to interference-noise (SINR) ratio is used to determine if an incoming signal can interfere or be received/captured. We use Two-Ray Ground radio propagation model, and the transmitting power is set so that the communication range is 115m, and the carrier sense threshold is set so that the default interference range is 250m. We use 2Mbps basic rate and 11Mbps data rate based on IEEE 802.11b. Each simulation runs *uplink* cbr flows with frame size 1024 bytes for 45 seconds. For each random experiment, we run the simulaiton for 20 times and obtain the average values unless otherwise stated.

4.1 Various Numbers of Clients

We first evaluate the performance of SMAC in a single cell by randomly placing varied numbers of clients in a 80m by 80m area with saturated offered load. The AP is placed in the center of the area. Figure 11 shows the total throughput for various numbers of clients in a cell. SMAC clearly scales when the number of clients increases and keeps the total throughput leveling at 6.2Mbps. On the other hand, the total throughput of 802.11 decreases more than 25% from 1 client to 128 clients in the cell.

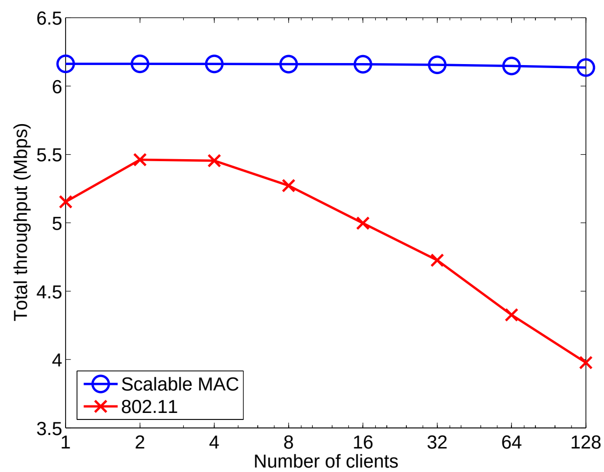


Figure 11: System throughput for varied number of clients in a cell

We further plot the throughput distribution of the 8, 32, and 128 clients contending in a BSS for 802.11 in Figure 12 and for SMAC in Figure 13. Although 802.11 have been thought to achieved long-term fairness among the clients, it does not possess this property even in this fairly long period of time (45 sec). The reasons are twofold. First, when the number of nodes is small, there are few collisions among the clients. As the number of nodes increases, however, the collided clients in the previous contention simply doubles its contention window and competes with other clients again with smaller channel attempt probability, resulting in short-term unfairness. Second, when capture effect comes into play, the nodes that are closer to the AP benefit from its stronger SINR ratio while the nodes that are farther from the AP suffers from the weaker SINR. When a collision happens, AP could capture the frame from the closer client and disregard the frame from the clients farther away, resulting in the intra-BSS unfairness. In fact, the more clients, the more severe the unfairness will be as shown in Figure 12. SMAC (Figure 13), on the other hand, relies on the schedule at the AP, and is able to keep all the flows sharing fair amount of wireless resource (max difference is within 20% of the highest throughput).

4.2 Various Offer Loads

It is well known that 802.11 provides good performance when the channel is under-utilized. However, the performance degrades when the available wireless resource becomes more and more scarce. We compare the total throughput of 802.11 and SMAC under various levels of offer loads.

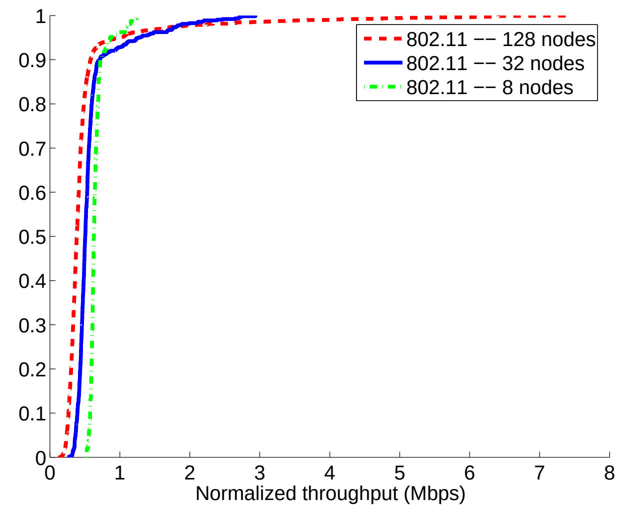


Figure 12: Throughput distribution for 802.11 for 8, 32, and 128 clients in a cell

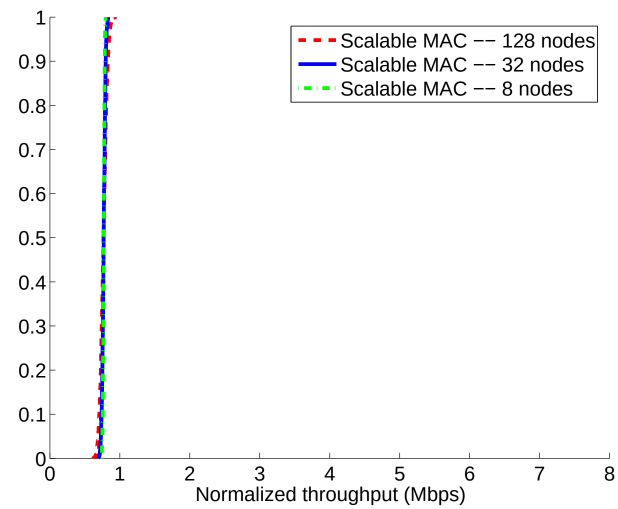


Figure 13: Throughput distribution for 802.11 for 8, 32, and 128 clients in a cell

We randomly place 8 clients in a 80 by 80 area with clients being applied offer load from 0.2 Mbps to 1.2 Mbps. With 8 clients the saturating point for 802.11 is around 0.8 Mbps as shown in Figure 14. SMAC is able to perform as efficient as 802.11 when the offer load is low (offer load less than 0.8 Mbps). When the offer load is high, it avoids the drawback of contention and achieves 20% more efficient channel utilization.

4.3 Channel Error Effect

In our previous simulations, a node fails to receive a frame only due to collisions. In reality, failure reception could also due to bad channel quality. We simulate the wireless channel for different bit error rate (BER) and compare the performance of SMAC and 802.11. Figure 15 shows the total throughput of 802.11 and SMAC for 32 clients in a BSS. Note that 802.11 does not distinguish failure transmission that is due to collision and bad channel quality. When

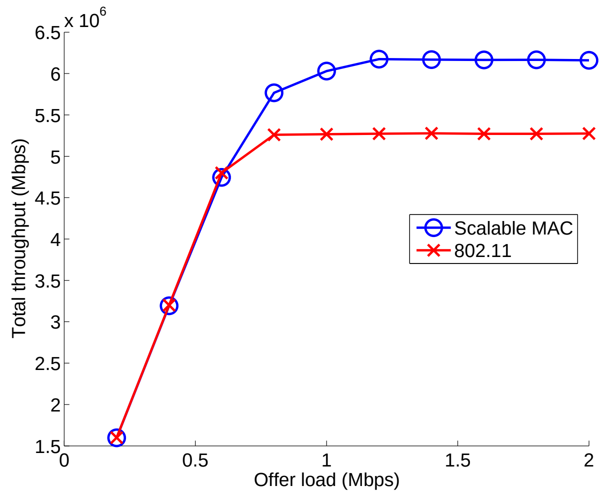


Figure 14: Average total throughput of 8 nodes for varied offer load

channel error happens, 802.11 clients simply doubles it contention window size before contending for the channel again. This is opposite to what it should have done - contend for the channel more aggresively. SMAC-enabled AP, on the other hand, quickly detects this error and reassign the optimal contention window and next client to transmit without incurring any unnecessary wasted time slots.

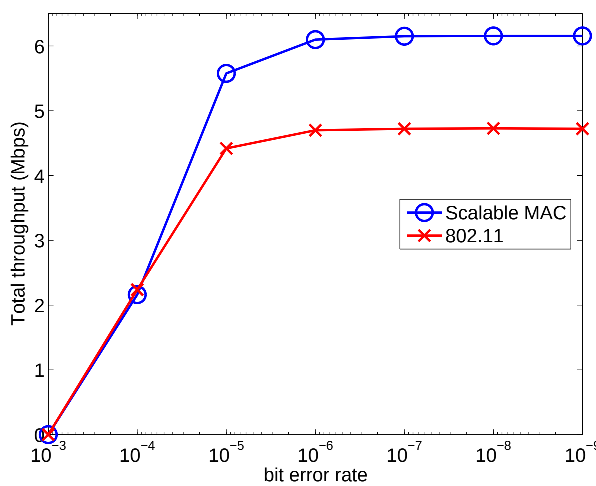


Figure 15: Average total throughput of 32 nodes for varied bit error rate (BER)

5. RELATED WORK

We review below the approaches to make MAC protocols to handle a lot of nodes more efficiently: spatial reuse [6, 8], multiple channels [1], DCF enhancements [3, 9], nodes grouping [11] and admission control [5].

Just like the cellular networks, a large area can be divided into many sub-areas and an AP can be assigned to each subarea. The less power the APs (and the nodes, of course) use, the smaller the cells are. Instead of placing a small number of APs, using many APs can reduce the number of nodes which is covered by an AP. To avoid inter-cell interference and hidden terminal problem, it is desirable for two adjacent cells to use different orthogonal channels, but it may not always be possible due to the limitation of the number of orthogonal channels.

Using multiple channels, the number of contending nodes can also be reduced [1]. If an AP has k interfaces each of which uses one of the k orthogonal channels and the nodes are uniformly assigned to one of the k channels, the number of contenders is reduced to $\frac{1}{k}$ of the number of all nodes. However, while this can mitigate the problem to some extent, it cannot be an ultimate solution because k is limited. In 802.11b, only three channels (1, 6 and 11) are orthogonal.

Many enhancements to DCF have been proposed [3, 9]. [3] shows that, if the number of contending nodes is known, the optimal CW can be calculated and it makes the sum of the collision time and idle time almost constant regardless of the number of nodes. This means the ideal DCF is scalable. Because it is impossible for the nodes to get the number of contenders, they use some approximation to make the protocol not scalable. [9] reduces CW into half after a successful transmission instead of resetting to CW_{min} and prevents rapid changes of CW value. It improves the performance of DCF, but does not five a solution to the scalability problem.

To reduce the number of nodes which contend at the same time, TMAC [11] has been proposed. In TMAC, all nodes are divided into g disjoint token-groups. The AP assigns a token to one token group to make the nodes in a specific token group to use the channel for some time. This is done in a coarse-time scale. If the token is assigned to one token group, the nodes in the token group contend using a DCF-like contention-based protocol. It also allows batch transmission in which nodes can send multiple packets after one success in contention and block ACK which acknowledges the reception of multiple packets using only one ACK packet.

If too many nodes are associated to an AP, the network connectivity of almost all nodes becomes unstable and scarce. This phenomenon is very common in large wireless networks. IQU [5] uses *admission control* to solve this problem. Every new node is queued at the AP before association. If the current channel is highly crowded, the AP does not allow the association of a new node. Instead, the AP queues the new node. Every associated node has a finite time slot during which it can access the network. After the timeout, it is put to the back of the queue and the connectivity is suspended. If the channel has some room for new nodes, the AP associates the top node in the queue to the network and enables the connectivity. One problem of this approach is that it changes the network service model. Because of the queue and timeout, the network connectivity is periodic and it makes the nodes to do all network-related jobs when they are connected. Definitely this approach causes problems when used with the upper-layer protocols such as TCP which are designed without consideration of this network access mode.

6. CONCLUSION

In this paper, we propose SMAC, a scalable MAC protocol, that preserves the advantages of 802.11 while avoiding its drawbacks. The design of SMAC leverages the efficient channel utilization of 802.11 DCF at low load and 802.11 PCF at high load. The SMAC-enabled AP controls the access of its clients by combining the contention and schedule-based access mechanisms and determining the optimal contention level among the clients. The SMAC-client simply indicates its intention of channel access. Through thorough analysis and evaluation, we have shown that SMAC preserves the channel utilization efficiency of 802.11 DCF, scales up to any number of nodes, and preserves perfect short-term and long-term fairness among the contending clients.

7. REFERENCES

[1] P. Bahl, R. Chandra, and J. Dunagan. Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 216–230, New York, NY, USA, 2004. ACM Press.

[2] F. Calí, M. Conti, and E. Gregori. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Transactions on Networking*, 8(6), December 2000.

[3] M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: An optimal access method for high throughput and fairness in rate diverse wireless lans. In *Proceedings of ACM SIGCOMM*, 2005.

[4] C. Hu and J. C. Hou. A reactive channel model for expediting wireless network simulation. In *ACM SIGMETRICS Poster*, 2005.

[5] A. P. Jardosh, K. Mittal, K. N. Ramachandran, E. M. Belding, and K. C. Almeroth. IQU: practical queue-based user association management for WLANs. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 158–169, New York, NY, USA, 2006. ACM Press.

[6] T.-S. Kim, H. Lim, and J. C. Hou. Improving Spatial Reuse through Tuning Transmit Power, Carrier Sense Threshold, and Data Rate in Multihop Wireless Networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, New York, NY, USA, 2006. ACM Press.

[7] Kodak easyshare-one zoom digital camera manual. `http://www.kodak.com/eknec/PageQuerier.jhtml?pq-path=6433&p`

[8] J. Monks, V. Bharghavan, and W. mei W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pages 219–228, 2001.

[9] Q. Ni, I. Aad, C. Barakat, and T. Turletti. Modeling and analysis of slow cw decrease ieee 802.11 wlan. In *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, 2003.

[10] J. Schiller. *Mobile communications*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, second edition, 2003.

[11] Y. Yuan and W. Arbaugh. Towards Scalable MAC Design for High-Speed Wireless LANs. Technical report, Department of Computer Science, University of Maryland, College Park, 2004.